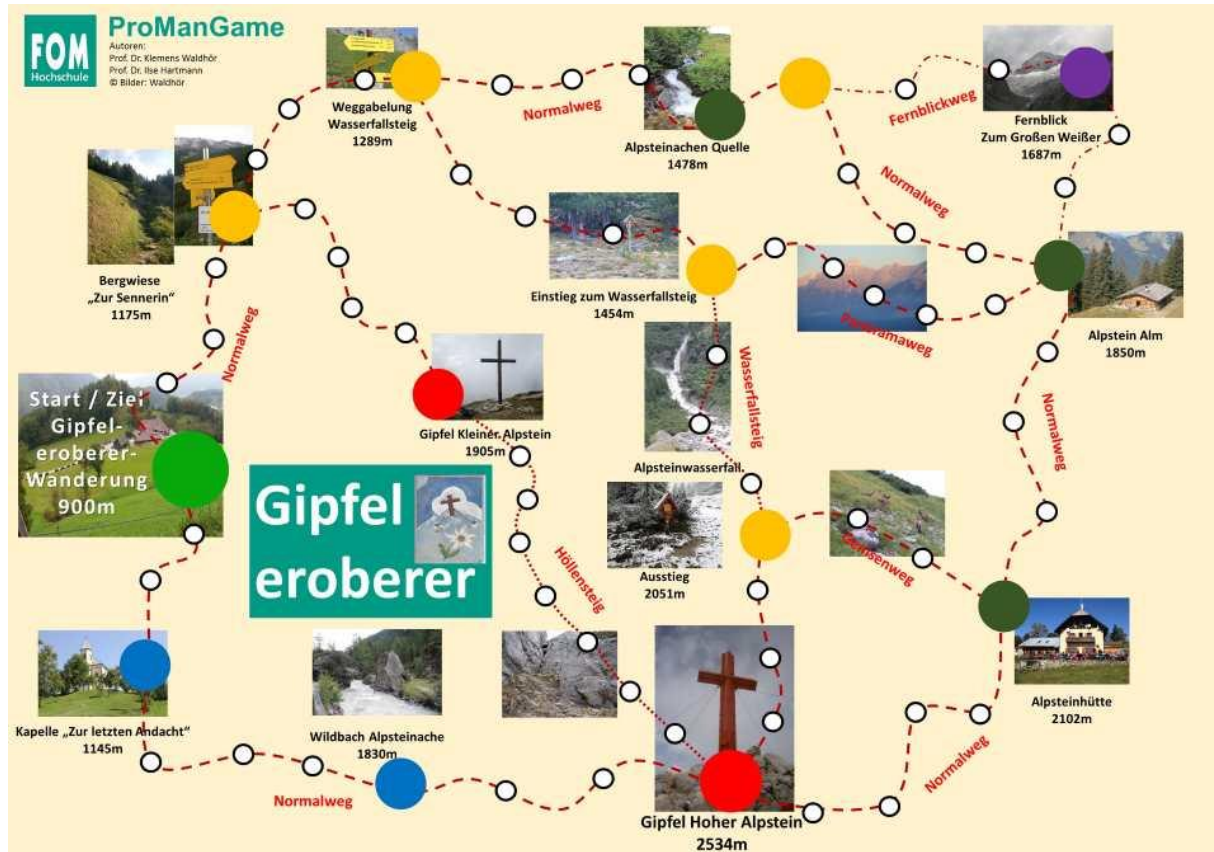


Projektaufgabe: Online Gipfelstürmer



Basis der Aufgaben ist der Gipfeleroberer-Spielplan. Die folgenden Aufgaben bilden nur einen Teil der Spielregeln ab.



Schwarz markierte Aufgaben sind Pflichtaufgaben, um diesen Teil positiv zu bestehen. **Blaue markierte Beschreibungen sind Bonusteile**, die die Note jeweils maximale Punktzahl treiben.

JavaScript Aufgaben



- Stellen Sie zwei **Spielfiguren** (Rot/Blau) auf dem Spielfeld. Die Darstellung und Implementierung (Methoden) soll dem Objektansatz von JavaScript folgen (sh. Vortrag Flath). Spielfiguren sollen ihren Zustand kennen (Position, aktuelles Feld auf Plan, Gabelung, Teamname, ...).
- Erstellen Sie ein Objekt, dass einen **Wegpunkt** darstellt. Wegpunkte sind entweder benannt () oder unbenannt (). Speichern Sie zu jedem Wegpunkt Typ, Koordinaten, Farbe, Name, Vorgänger, Nachfolger Wegpunkt und Weg, auf dem er sich befindet. Jeder Wegpunkt soll eine eindeutige Identifizierung haben).

3. Schreiben Sie eine Methode **figurPositionieren**, um eine Spielfigur an eine bestimmte Stelle des Spielfeldes zu positionieren (mittels x, y Koordinaten).
4. Erweitern Sie **figurPositionieren** derart, dass falls sich durch Positionieren einer Spielfigur eine schon an dieser Position platzierte Figur überdecken werden sollte, die aktuell zu positionierende Spielfigur neben der ursprünglichen positionierten Spielfigur platziert wird. Alternativ können Sie auch beide so verschieben, dass sie sich nicht überdecken.
5. Schreiben Sie eine Methode **figurPositionierenAnimiert**, um eine Figur von einer an eine bestimmte Stelle des Spielfeldes **animiert** zu bewegen.
6. Schreiben Sie eine Methode **figurWegpunktPositionieren** (basierend auf **figurPositionieren**) um eine Spielfigur auf einem bestimmten Wegpunkt des Spielfeldes zu positionieren. Dies soll entweder über den Wegpunktnamen oder seine eindeutige Identifizierung möglich sein.
7. Erstellen Sie einen Knopf, mit dem die Spielfiguren an den Start mittels der Methode **figurAufStartPositionieren** positioniert werden können.
8. Erstellen Sie zwei **Würfel** (rot / blau)  mit den möglichen Wurfzahlen 1-6. Die Darstellung und Implementierung (Methoden) der Würfel soll dem Objektansatz von JavaScript folgen. Würfel soll seinen Zustand kennen (aktuelle Anzeige, darf gedrückt werden, Farbe, ...)
9. Implementieren Sie einen Würfel mit beliebiger Wurfzahlen 1-n.
10. Implementieren Sie eine Methode **wuerfeln**, die als Ergebnis eine Zufallszahl von 1- 6 (bzw. 1-n) liefert. Der Würfel zeigt die entsprechend gewürfelte Zahl an.
11. Der Würfel muss diese Zahl als Punkte anzeigen wie im Bild des Würfels.
12. Implementieren Sie eine Methode **figurZiehen** um eine Figur vom aktuellen Wegpunkt zum um die gewürfelte Zahl passenden nächsten Wegpunkt zu bewegen (einen Spielzug durchführen).
13. Schreiben Sie eine weitere Methode **wuerfelnMitZiehen** basierend auf der schon durchgeführten Implementierung **wuerfeln**, die die der Würfelfarbe entsprechende Spielfigur mit der Methode aus **figurZiehen** bewegt, wenn auf den farblich passenden Würfel geklickt wird.
14. Kommt die Spielfigur während eines Spielzuges über einen Wegpunkt, der eine Weggabelung darstellt, wird an der Weggabelung gestoppt. Über den Zielwegpunkt hinaus darf nicht gezogen werden.
15. Es soll immer nur abwechselnd gewürfelt werden können.
16. Die einzelnen Spielzüge sollen gespeichert werden. Beim nächsten Aufruf soll die letzte Spielsituation wiederhergestellt werden können.
17. Implementieren Sie die Aufgabe für mehr als zwei Spielfiguren und Würfel.
18. Sie können weitere eigene Ideen realisieren.

Lösungshinweise: z-index (css Attribut) ermöglicht Elemente übereinander zu positionieren. Verwenden Sie das Storage-Objekt zum persistenten Speichern von Informationen. **Überlegen Sie den Einsatz von jQuery.**

Führen Sie die Implementierung so durch, dass weitere Aspekte des Gipfelstürmerspiels von zukünftigen Gruppen implementiert werden können.

Überlegen Sie eine Aufteilung der Aufgaben in der Gruppe. Was kann unabhängig voneinander implementiert werden?

Dokumentation der praktischen Projektaufgabe:

Für die Projektaufgabe Webprogrammierung soll die Dokumentation folgendes umfassen (alles in einer Zip-Datei, falls für das Hochladen zu groß, bitte auf CD abgeben). Bitte für die unterschiedlichen Aufgaben jeweils eigene Ordner mit Aufgabennummer verwenden:

- Beschreibung der Software
- Schematische Architekturbeschreibung
- Software/Hardwarevoraussetzungen
- Installationsanweisungen
- Benutzungshinweise
- Liste mit allen Dateien (je nach Aufgabe)
- Source Code (nur elektronisch)
- In den Programmdateien Verwendung von Kommentaren, insbesondere sollte jede Programmdatei einen Headerkommentar enthalten mit Informationen über die Funktion der Datei und die Autoren, Rechtehinweis.
- Readme.html
- „Bonusmaterial“ zum Einsatz der Projektanwendung, wie z.B. Video, Bildschirm-Hardcopies o.ä.

Rechte an der Software:

Open Source (etwa wie bei Apache Foundation), es muss zumindest gewährleistet sein, dass die Ergebnisse im Rahmen der FOM eingesetzt werden können.

Die Übungsaufgaben sollen jeweils in einer Gruppe (2 – 4 Studenten) gemeinsam erbracht werden.